Converting to R3.14: Channel Access Clients

kasemir@lanl.gov

Experience based on...

- A simple 'caget' program.
- ChannelArchiver's sampling Engine.

'caget': No CA-related Changes

- Changes to the build environment:
  - New EPICS_HOST_ARCH variable.
  - “Db” library replaced by “dbStaticHost” et al.
  - Need C++ - aware linker even for pure C code because of EPICS base library.
- No more tsStampToText
  - It's been moved to an extension.
  - Do-it-yourself:

```
epicsTimeToTM(&tm, &nano, &time_stamp);
printf("%02d/%02d/%04d ....",
       tm.tm_mon + 1, tm.tm_mday, ...);
```

'ArchiveEngine': New CA Features

- ca_context_create
  - Replaces ca_task_initialize.
  - Allows to pick multi-threaded callbacks.
- ca_current_context, ca_attach_context
  - Allows usage of same CA context across multiple user threads.
- ca_context_destroy
  - Replaces ca_task_exit.
- no more ca_pend_io, ca_poll, ...

More New CA Features

- ca_create_channel
  - Replaces ca_search_and_connect.
  - Now with configurable priority.
- ca_create_subscription
  - Replaces ca_add_..._event.
- ca_clear_subscription
  - Replaces ca_clear_event.

Multi-Threading

- Enabling preemptive CA callbacks is much easier than dealing with the consequences.
  - Alan Cox: "A Computer is a state machine. Threads are for people who can't program state machines."
  - Add semaphores. Jeff Hill's idea of a 'guard' class helps:
    - Takes & releases semaphore in constructor/destr.
    - Establish, document & check lock hierarchy by passing C++ reference to all methods that need the semaphore.

Lessons Learned: Locking user code vs. CA

- “Callback lock” in CA client library
  - Taken in many CA API routines to prohibit callbacks while e.g. a channel gets deleted.
  - Taken inside CA callbacks.
- This lead to deadlock:
  - Engine locked itself, called ca_clear_channel, which takes callback lock.
  - CA invoked subscription callback *for a different channel* (callback lock taken),
    which then locked Engine to update status.

CA R3.14 Summary

- Fully backwards compatible
  - Any issues are outside of CA.
- New API is easy enough to understand
  - Use of a 'guard' class would help w/ regard to locking issues between multi-threaded user code and CA library.